UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/943,563 | 08/30/2001 | Maria-Jose Arbulu Barturen | FR920010023US1 | 2529 |

45832 7590 09/24/2009

DILLON & YUDELL LLP
8911 N. CAPITAL OF TEXAS HWY.,
SUITE 2110
AUSTIN, TX 78759

| EXAMINER | |
|---|---|
| PHAM, CHRYSTINE | |
| ART UNIT | PAPER NUMBER |
| 2192 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 09/24/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

---

*Ex parte* MARIA-JOSE ARBULU BARTUREN,
MARIANO DIEZ FERNANDEZ,
IGNACIO FERNANDEZ GONZALEZ,
and ELISA MARTIN GARIJO

---

Appeal 2008-005422
Application 09/943,563[1]
Technology Center 2100

---

Decided: September 24, 2009

---

Before LEE E. BARRETT, LANCE LEONARD BARRY, and HOWARD B. BLANKENSHIP, *Administrative Patent Judges*.

BARRETT, *Administrative Patent Judge*.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) from the final rejection of claims 1-17. We have jurisdiction pursuant to 35 U.S.C. § 6(b).

We affirm-in-part.

---

[1] Filed August 30, 2001, titled "Integrated System and Method for the Management of a Complete End-to-End Software Delivery Process." The real party in interest is International Business Machines Corporation. Corrected Brief filed February 21, 2007 (Br.), 1.

## STATEMENT OF THE CASE

*The invention*

The present invention generally relates to the field of software delivery, and in particular to an integrated system and to a method to completely manage an end-to-end software delivery process, adapted to manage a software product along the whole life cycle thereof, from development to installation in production. Spec. 1, ll. 7-12.

*The claims*

Claim 1 is reproduced below:

> 1. An integrated data processing system for managing a process of delivery of software products to target software product execution units in a network environment, comprising:
>
> a central repository for storing software components of at least one software product;
>
> a first sub-system for identifying within the central repository software components of a software product to be delivered;
>
> a second sub-system for creating at least one software product package from the identified software components identified by the first sub-system, and
>
> a third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units.

*The references*

| | | |
|---|---|---|
| Apfel | US 5,974,454 | Oct. 26, 1999 |
| Albright | US 6,110,228 | Aug. 29, 2000 |
| Goiffon | US 6,427,230 B1 | Jul. 30, 2002 (filed Nov. 9, 1998) |

*The rejections*

Claims 1, 2, 4-8, 10, 12-15, and 17 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Goiffon.

Claims 3 and 9 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Goiffon and Apfel.

Claims 11 and 16 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Goiffon and Albright.

## PRINCIPLES OF LAW

*Anticipation*

"Anticipation requires the presence in a single prior art disclosure of all elements of a claimed invention arranged as in the claim." *Connell v. Sears, Roebuck & Co.*, 722 F.2d 1542, 1548 (Fed. Cir. 1983).

*Obviousness*

Obviousness requires that the combination of references teach or suggest to a person of ordinary skill in the art all of the claim limitations. 35 U.S.C. § 103(a).

*Arguments not made are waived*

Arguments not made are considered waived. *Cf. In re Baxter Travenol Labs.*, 952 F.2d 388, 391 (Fed. Cir. 1991) ("It is not the function of this court to examine the claims in greater detail than argued by an appellant, looking for nonobvious distinctions over the prior art."); *In re Wiechert*, 370 F.2d 927, 936 (CCPA 1967) ("This court has uniformly followed the

sound rule that an issue raised below which is not argued in this court, even if it has been properly brought here by a reason of appeal, is regarded as abandoned and will not be considered. It is our function as a court to decide disputed issues, not to create them."); *In re Wiseman*, 596 F.2d 1019, 1022 (CCPA 1979) (arguments must first be presented to the Board before they can be argued on appeal).

## DISCUSSION

### *I. Anticipation*

#### *A. Claims 1, 4-6, and 8*

Claims 1, 4-6, and 8 are grouped to stand or fall together.

Appellants argue with respect to claim 1 that the Examiner-cited portions of Goiffon do not disclose: (1) a "system for managing a process of delivery of software products" as recited in the preamble (Br.[2] 12-14); (2) a "central repository for storing software components of at least one software product" (Br. 14-15); and (3) a "third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units" (Br. 15-16).

We note that Appellants do not deny that Goiffon may teach these three limitations, and, in fact, candidly admit that Goiffon describes the first limitation, but argue that the Examiner erred in finding that the cited

[2] We refer to Corrected Brief filed February 21, 2007 (Br.) and the Corrected Reply Brief filed July 10, 2007 (Reply Br.).

portions of Goiffon describe the limitations. We agree with Appellants that the Examiner's findings, as stated and understood, are in error. Appellants are not responsible for making up their own rejection. Nevertheless, it appears that other portions of Goiffon support the anticipation rejection and accordingly the rejection is affirmed to expedite prosecution. Because anticipation is a factual inquiry and because it appears that Appellants are aware of the relevant teachings of Goiffon, due process requirements appear to be satisfied and we do not label this a new ground of rejection.

As an introduction, Appellants argue that the Examiner points to numerous different components of Goiffon as allegedly disclosing each of the limitations of claim 1, but argue that "[t]he basis for contending that these disparate components allegedly disclose the invention of Claim 1 is far from clear." Br. 11. Appellants note, for example, that the Examiner points to eight different components of Goiffon as corresponding to the "central repository" of claim 1, but "does not even attempt to explain what combination of the eight (8) separate items enumerated above allegedly constitutes the 'central repository for storing software components of at least one software product' of Claim 1." *Id.* at 12. It is argued that "[t]he Final Action similarly cites to a laundry list of components from Goiffon as disclosing each of the first through third sub-systems recited [sic, in] Claim 1. *Id.*

We agree with Appellants that the rejection is not clear. Ideally, an anticipation rejection should map the limitations of the claim one-by-one onto a specific element or function in the prior art reference. The

Examiner's finding that the limitation "central repository for storing software components of at least one software product" corresponds to "at least [1] *object repository*, [2] *software constructs*, [3] *packages* Abstract; [4] *AIM Server 214*, [5] *Element Repository 220* Fig.2B & associated text; col. 12:7-15; col. 12:23-67; [6] *Host A 228*, [7] *Memory 229* Fig.2B & associated text; *Host A 228, Memory 229,* [8] *data modules* col. 12:57-co1.13:20," Ans. 4-5 (numbers in brackets added), is not helpful because it cites many different elements for the same limitation. This gives the appearance that the Examiner is citing everything in the hopes that something might stick. Nevertheless, we make our decision on the factual teachings of Goiffon.

The three limitations at issue are discussed below.

1. A "system for managing a process of delivery of software products"

Appellants argue that the Examiner-cited portions of Goiffon refer to an Element Inventory 102 and an "export" operation, which do not disclose a "system for managing a process of delivery of software products" as recited in the preamble of claim 1. Br. 12-14. It is argued that that Element Inventory 102 stores objects or elements that are used to manage the code and data components and the objects store meta-data about data or code residing elsewhere. *Id.* at 12-13. It is argued that "Goiffon expressly states that the actual code and data modules are stored in Host A 228, which is a separate element." *Id.* at 13. Appellants argue that "the 'export function' . . . of Goiffon clearly does not involve the delivery of a software product to an

execution unit, but instead involves the export of a meta-data object that points to, and describes, a software object that may be used to perform a function." *Id.* "Appellants do not dispute that Goiffon discloses that codes and data modules may be grouped together to form a package, and that such a package of actual software modules can be 'migrated' to a new platform. (*See*, e.g., Goiffon at Col. 3, lines 27-32 and Col. 4, lines 15-26)." Br. 14. (Appellants' candor is noted and appreciated.) However, it is argued, "these 'packages' indisputably are not stored in the Element Inventory 102, and have nothing to do with the element 'export function of Goiffon that forms the basis for the pending rejection of Claim 1. . . . The Examiner cannot overcome this by improperly mixing and matching descriptions of what is done with the actual code and data modules and the meta-data objects as has been done in the pending rejections." *Id.*

The Examiner notes that Appellants acknowledge that the packages in Goiffon contain actual software modules and are distributed (migrated) to a new platform. Ans. 15. The Examiner finds that Goiffon teaches a package "element" is stored in the Element Inventory 102 via a "create element" service call, and "that since the 'element' to be stored in the Element Inventory is of type *package*, it includes the actual software code/modules (as opposed to just objects or metadata describing the software code/modules) which make up the *package*." *Id.* The Examiner states that the "export" function delivers a copy of an element to a remote system and the "import" function receives a copy of an element and stores the copy in the Element Inventory 102, which is "centralized." *Id.* at 16. The Examiner

states that "[i]t should be understood also that when an 'element' of type *package* is being exported (i.e., distributed and/or delivered) to a remote system, it contains the actual software code/modules as well as elements (i.e., metadata) describing the actual software code/modules." *Id.*

Appellants respond that Goiffon expressly, and repeatedly, states that the code and data modules are not stored in the Element Inventory 102. It is argued that a package element is merely a group of meta-data elements that model the code and data modules that Goiffon repeatedly states are stored elsewhere. Reply Br. 1-8.

We agree with Appellants that Element Inventory 102 does not store software components and that the "export" operation does not export software products. The Element Inventory 102 is a collection of elements, each of which is an object storing meta-data about other code, data or system entities residing elsewhere. Col. 12, ll. 57-64. The Examiner appears to confuse "packages" of code and data modules (Goiffon, col. 2, ll. 36-37) with "Element Packages." "[A]n Element Package will comprise elements that represent, and model, all code and data modules that are needed to perform one or more predetermined functions." Col. 22, ll. 31-34. That is, an Element Package is meta-data that models code and data modules, not the code and data modules themselves. The "export" operation exports elements, not actual software code. Thus, we agree that the Examiner has not pointed to the correct portions of Goiffon.

However, Appellants candidly admit that Goiffon teaches a "system for managing a process of delivery of software products," for example, at

column 3, lines 27-32 and column 4, lines 15-26, and we agree we this finding. Thus, we find that Goiffon teaches the limitation despite the error in the Examiner's reasoning.

2. A "central repository for storing software components of at least one software product"

Appellants argue that the Examiner-cited portions of Goiffon do not disclose a "central repository for storing software components of at least one software product." Br. 14-15. It is noted that the Examiner points to eight different components of Goiffon as corresponding to the "central repository" of claim 1, but "does not even attempt to explain what combination of the eight (8) separate items enumerated above allegedly constitutes the 'central repository for storing software components of at least one software product' of Claim 1." *Id.* at 12. Appellants understand that the Examiner takes the position that the Element Inventory 102 that is part of the Element Repository 220 of Figure 2B of Goiffon is the "central repository." *Id.* at 14. Appellants note that Goiffon teaches at column 6, lines 58-66 that the Element Inventory 102 stores meta-data, not actual code, which is stored elsewhere. *Id.* at 15. Appellants argue that "[w]hile Appellants do not dispute that Goiffon discusses creating packages of a plurality of code and data modules that may be used to perform a given task (*see* Goiffon at Col. 4, lines 15-26), such code and data modules are not what is stored in the Element Inventory/Repository." *Id.*

The Examiner persists in arguing that the Element Inventory 102 stores software code and concludes: "Thus, contrary to Appellants' argument, Element Inventory 102 clearly anticipates a central repository for storing 'reusable' software components (i.e., packages or software code/modules) of a least one software product (i.e., package)." Ans. 17.

As stated in the discussion of limitation (1), the Element Inventory 102 does not store software code. Nevertheless, Goiffon describes in connection with Figure 2B that "Host A 228 includes Memory 229 which stores code and data modules." Col. 13, ll. 6-7. Thus, although the Examiner erred in finding that Element Inventory 102 corresponds to the claimed "central repository for storing software components of at least one software product," we find the limitation is met by memory 229.

3. A "third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units"

Appellants lastly argue that the Examiner-cited portions of Goiffon do not disclose a "third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units." *Id.* at 15-16. It is argued that the "export" operation relied upon by the Examiner provides copies of elements not code, as claimed. *Id.* at 15.

The Examiner states that Goiffon teaches wrapping packages with layers of software called a "wrapper" and that "[i]t is clear that the wrapping the created package anticipates preparing the package for distribution to

different remote system 107." Ans. 18. The Examiner refers back to previous discussion of exporting packages in the Element Inventory 102. *Id.*

We do not agree with the Examiner that the Element Packages exported from the Element Inventory 102 are software code. As previously discussed, an Element Package is meta-data that models software code and the Element Inventory 102 stores only meta-data. Nevertheless, as noted in the discussion of a "system for managing a process of delivery of software products," Goiffon teaches that packages containing code and data modules are migrated to a new platform at, for example, column 3, lines 27-32 and column 4, lines 15-26. Thus, there must be a "third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units." The closest identifiable structure is probably the Host A 228. Therefore, in spite of the Examiner's statements, we find that Goiffon teaches "third sub-system for distributing the at least one software product package created by the second sub-system to the target software product execution units."

*Conclusion*

Because we find that Goiffon teaches the three argued limitations, albeit at different locations than cited by the Examiner, the anticipation rejection of claim 1 is affirmed. Since claims 1, 4-6, and 8 are grouped to stand or fall together, the rejection of claims 4-6 and 8 is also affirmed.

*B. Claims 2 and 10*

Appellants argue that the Examiner-cited portions of Goiffon do not describe a "software package distribution repository" as recited in claim 2 and a similar limitation in claim 10. Br. 16. The Examiner cites to element 1024 of Figure 10 and elements 1808, 1816, and 1828 of Figures 18A and 18B. Final Rej. 6. Appellants argue that the cited portions are directed to processing steps that are used to create "Element Packages," which are nothing more than meta-data objects, and these meta-data objects are not created from an identified group of software components that are stored in a central repository. Br. 16.

The Examiner finds that the "Element Inventory 102 clearly anticipates a 'central distribution repository for storing at least one software product package' . . . created by the second sub-system." Ans. 19.

As discussed in connection with claim 1, we agree with Appellants that the Element Inventory 102 does not store software code. However, we found that software code is stored in memory 229 of Host A 228. Goiffon describes that packages of code and data modules are created and distributed (e.g., col. 3, ll. 27-32 and col. 4, ll. 15-26) and these packages have to be stored somewhere while awaiting distribution. Accordingly, we affirm the rejection of claims 2 and 10.

*C. Claim 7*

Appellants argue that the Examiner-cited portions of Goiffon do not describe a "sixth subsystem for recording information provided by at least

one of the first through fifth sub-systems of the integrated data processing system during delivery of the software product." Br. 16-17.

The Examiner finds that line 227 of Figure 2A and line 240 of Figure 2B meet the limitation because the Export Elements service reads elements from the Element Inventory 102 and writes them into a file as indicated by dashed line 240, where the writing is the recording, and the Import Elements service reads elements from a file and writes them into the Element Inventory 102. Ans. 19-20.

Appellants argue that "this argument confuses reading and writing the elements between Client Server 216 and Data Processing System 219 with recording information provided by various subsystems." Reply Br. 8. It is argued that the rejection takes "the clearly inconsistent position that importing and exporting elements comprises both the distribution of the software package by the third sub-system and the recoding [sic] of information by the sixth sub-system." *Id.* at 9.

We agree with Appellants that Goiffon does not teach recording information provided by the various sub-systems. For example, in a "first sub-system for identifying . . . software components of a software product to be delivered," the information provided is the identified software components, not the actual software components. In addition, we agree that reading and writing "elements," which are not software components, does not meet the "recording information" limitation. The rejection of claim 7 is reversed.

*D. Claims 12-15 and 17*

Appellants argue, *inter alia*, that the Examiner-cited portions of Goiffon do not describe "storing the built software product in the central repository" and storing both the components and the built software product "in the central repository" as recited in claim 12. Br. 19-20. The Examiner relies on the same portions of Goiffon as for claim 2. Appellants argue that the Element Inventory 102 in Goiffon does not store software components and that the stored Element Packages are not "built software products," but are a meta-data objects. *Id.* at 19.

As previously discussed, we agree with Appellants that the Element Inventory 102 does not store software components and the Element Packages stored in the inventory 102 are meta-data objects, not software code. The Element Inventory 102 likewise does not store "built software products." Goiffon describes that software products (packages containing code and data modules) are built and distributed, and therefore must be stored somewhere before distribution. However, claim 12, unlike claim 2, requires the built software products to be stored in the same central repository as the software components. While it is possible, and maybe even likely, that the packages created in Goiffon are stored in memory 229 of Host A 228 with the code and data modules, this is not necessarily inherent as required to anticipate. Accordingly, we reverse the rejection of claims 12-15 and 17.

*II. Obviousness*

*A. Claims 3 and 9*

Claim 3 recites "the third sub-system distributes the at least one software product package to target software product execution units belonging to at least one environment according to at least one role assigned to the at least one software product package by the second sub-system." Claim 9 contains a similar limitation.

The Examiner finds that Goiffon teaches distributing a "software product package to target software product execution units belonging to at least one environment (see at least *different* operating environment col. 8:58-67)." Final Rej. 11. However, the Examiner finds that Goiffon does not describe distributing software "according to at least one role assigned to the at least one software product package by the second sub-system." The Examiner finds that "*Apfel* teaches assigning (i.e., associating) each software product package with a role (i.e., environment or operating system) and distributing said package to target software product execution units belonging to an environment according (i.e., matching) said role (e.g., . . . col. 6:65-67 . . . col. 9:35-42)." *Id.* The Examiner states that Goiffon and Apfel are analogous art because they are directed to distributing software packages. *Id.*

Appellants dispute that Apfel discloses the recitation of claims 3 and 9, but do not argue this point as they contend that one of ordinary skill in the art would not have been motivated to combine Goiffon and Apfel in the manner suggested by the Examiner. Br. 21. It is argued that Goiffon is

directed to a system for managing reusable groups of software while Apfel is directed to a method and system for updating software programs that are already resident on target computers, so the disclosures have nothing in common, and are directed to entirely different problems and solutions. *Id.*

A "role" may be, for example, an operating system. *See* Spec. 22, ll. 14-15 ("For instance, a target system role may be Windows NT Server, MQ Series client, SAP NT Server, etc.").

Goiffon describes creating code "wrappers" for code modules to allow code reuse by new platform architectures. For example:

> Using the interface definition provided by an Element Package, "wrapper" code can be developed that translates the Element Package interface defined by the member elements to a different interface technology. That is, the wrapper provides an interface that allows the member code and data modules to be accessed from an environment other than the one in which the member modules operate.

Col. 22, ll. 45-52. We find that Goiffon alone teaches creating software product packages with the code "wrappers" assigned by the system which are distributed to target software product execution units based on their environment. In addition, we agree with the Examiner's finding that Apfel describes distributing software product packages based on the operating system environment and that it would have been obvious to create and distribute different software packages based on target operating systems in Goiffon (if this was not already suggested) to increase the code reuse. Appellants' arguments that Apfel is not combinable because it is directed to updating programs already resident on target computers is not persuasive

because Apfel describes "installing" and "updating" program module components, e.g., column 6, lines 27-28. Accordingly, we are not persuaded of error in the obviousness rejection. The rejection of claims 3 and 9 is affirmed.

*B. Claims 11 and 16*

The Examiner cites Albright for the limitations of claims 11 and 16. Final Rej. 12-13.

Appellants argue that Albright is not recited as disclosing any of the limitations of claims 8 and 12, from which claims 11 and 16 depend, and thus claims 11 and 16 are patentable because claims 8 and 12 are patentable. Br. 21-22.

The Examiner responds that Albright is not relied upon for the limitations of claims 8 and 12. Ans. 22.

Appellants have not argued any error in the Examiner's rejection of claim 11 if the rejection of claim 8 is affirmed. Thus, the rejection of claim 11 is affirmed. Since the Examiner does not rely on Albright for claim 12, the rejection of claim 16 is reversed because claim 12 is reversed.

## CONCLUSION

The rejection of claims 1, 2, 4-8, and 10 under 35 U.S.C. § 102(e) over Goiffon is affirmed. The rejection of claims 12-15 and 17 under 35 U.S.C. § 102(e) over Goiffon is reversed.

The rejection of claims 3 and 9 under 35 U.S.C. § 103(a) over Goiffon and Apfel is affirmed.

The rejection of claim 11 under 35 U.S.C. § 103(a) over Goiffon and Albright is affirmed. The rejection of claim 16 under 35 U.S.C. § 103(a) over Goiffon and Albright is reversed.

Requests for extensions of time are governed by 37 C.F.R. § 1.136(b). *See* 37 C.F.R. § 41.50(f).

AFFIRMED-IN-PART

rwk

DILLON & YUDELL LLP
8911 N. CAPITAL OF TEXAS HWY.,
SUITE 2110
AUSTIN, TX 78759